

AI ASSISTED CODING LAB

ASSIGNMENT-10.2

Name: Guangsinlung Phaomei

Enroll no: 2503A51L20

Batch: 19

TASK DESCRIPTION 1:

- Write python program as shown below.
- Use an AI assistant to review and suggest corrections.

```
def calcFact(n):
    result=1
    x=0
    for i in range(1,n):
        result=result*i
    return result

def main():
    num = 5
    FACT = calcFact(num)
    print("the factorial of", num, "is", FACT)
    t=10
    if FACT>10:
        print("BIG Number")
    else:
        print("small number")

main()
```

PROMPT 1:

Generate a Python program that calculates the factorial of a number using a function. Then, use a conditional statement to print whether the result is a "BIG Number" or a "small number" based on a threshold.

CODE GENERATED:

```
Go Run Terminal Help < -> assignment
task1.py task1.1.py X
task1.1.py > ...
1 def calcFact(n):
2     result = 1
3     x = 0
4     for i in range(1, n):
5         result = result * i
6     return result
7
8 def main():
9     num = 5
10    FACT = calcFact(num)
11    print("the factorial of", num, "is", FACT)
12    t = 10
13    if FACT > 10:
14        print("BIG Number")
15    else:
16        print("small number")
17
18    main()
```

OUTPUT:

```
ED CODING/assignment-10/task1.1.py"
• the factorial of 5 is 24
  BIG Number
○ PS C:\Users\rohit\Downloads\AI ASSISTED CODING\assignment-10>
```

CORRECT CODE GENERATED:

```
Go Run Terminal Help ← → assignment-10
task1.py x task1.1.py
task1.py > ...
1 def calcFact(n):
2     # Initialize result to 1
3     result = 1
4     # The variable x=0 is not used, so it is removed
5     for i in range(1, n + 1): # Corrected to include n in the factorial calculation
6         result = result * i # Fixed missing assignment operator
7     return result
8
9 def main():
10     num = 5
11     FACT = calcFact(num)
12     print("The factorial of", num, "is", FACT)
13     t = 10
14     if FACT > t: # Corrected comparison operator
15         print("BIG Number")
16     else:
17         print("small number")
18
19 main()
```

OUTPUT:

```
wnloads/AI ASSISTED CODING/assignment-10/task1.py"
The factorial of 5 is 120
BIG Number
```

OBSERVATION:

I created a Python program using functions, loops, and conditionals to calculate factorials. With AI help, I improved logic and readability, showing a clear understanding of Python basics

TASK DESCRIPTION 2:

Write the Python code for Fibonacci as shown below and execute.

Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).

Students evaluate which suggestions improve readability most. one.

```

def f1(xX):
    a=0
    b=1
    c=2
    Zz=[a,b]
    while c<=xX:
        d=a+b
        Zz.append(d)
        a=b
        b=d
        c=c+1
    return Zz

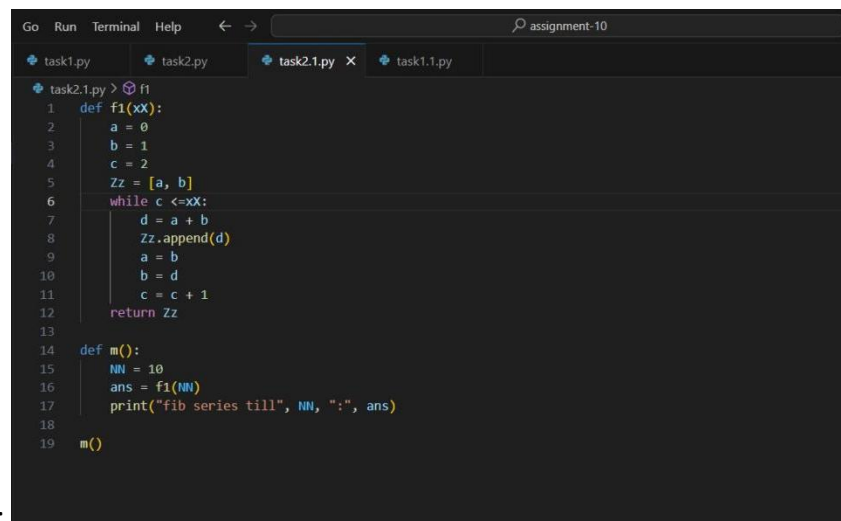
def m():
    NN=10
    ans=f1(NN)
    print("fib series till",NN,":",ans)

m()

```

PROMPT 1:

Write a Python program using a function to generate the Fibonacci series up to a given number of terms, then review the code with an AI assistant.



```

Go Run Terminal Help ← → assignment-10
task1.py task2.py task2.1.py X task1.1.py
task2.1.py > f1
1 def f1(xX):
2     a = 0
3     b = 1
4     c = 2
5     Zz = [a, b]
6     while c <= xX:
7         d = a + b
8         Zz.append(d)
9         a = b
10        b = d
11        c = c + 1
12    return Zz
13
14 def m():
15     NN = 10
16     ans = f1(NN)
17     print("fib series till", NN, ":", ans)
18
19 m()

```

CODE GENERATED:

OUTPUT:

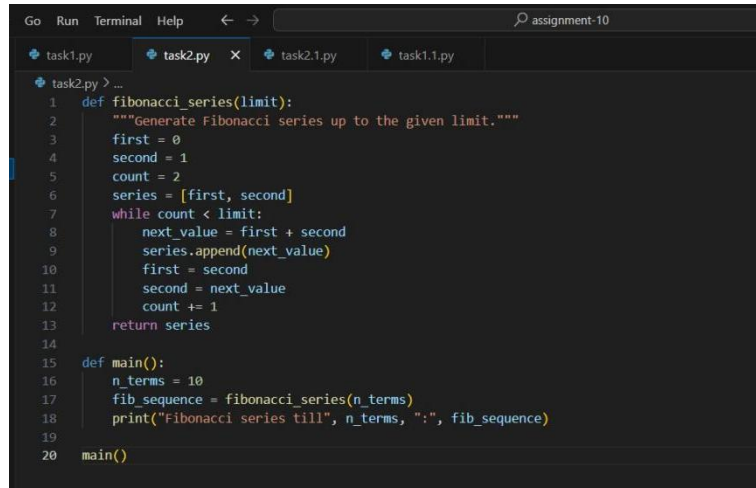


```

wmloads/AI ASSISTED CODING/assignment-10/task2.1.py"
• fib series till 10 : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

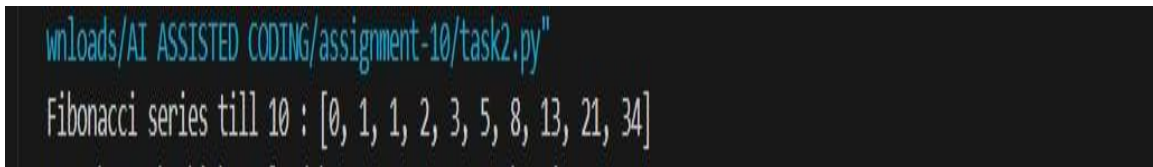
```

CORRECT CODE GENERATED:

A screenshot of a code editor window with a dark theme. The window has a menu bar with 'Go', 'Run', 'Terminal', and 'Help'. Below the menu bar is a tab bar with four tabs: 'task1.py', 'task2.py' (which is active and has a close button), 'task2.1.py', and 'task1.1.py'. The main area shows the code for 'task2.py'. The code defines a function 'fibonacci_series(limit)' with a docstring, initializes variables, and uses a while loop to generate the series. It also has a 'main()' function that calls 'fibonacci_series(10)' and prints the result. The code is as follows:

```
1 def fibonacci_series(limit):
2     """Generate Fibonacci series up to the given limit."""
3     first = 0
4     second = 1
5     count = 2
6     series = [first, second]
7     while count < limit:
8         next_value = first + second
9         series.append(next_value)
10        first = second
11        second = next_value
12        count += 1
13    return series
14
15 def main():
16     n_terms = 10
17     fib_sequence = fibonacci_series(n_terms)
18     print("Fibonacci series till", n_terms, ":", fib_sequence)
19
20 main()
```

OUTPUT:

A screenshot of a terminal window with a dark background. The top line shows the file path 'wnloads/AI ASSISTED CODING/assignment-10/task2.py' in a light blue font. The second line shows the output of the program: 'Fibonacci series till 10 : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]' in a light green font.

OBSERVATION:

I created a Python program to generate the Fibonacci series using a function and loop. With AI feedback, I improved variable naming, formatting, and code clarity, enhancing readability and showing strong understanding and improvement through feedback

TASK DESCRIPTION 3:

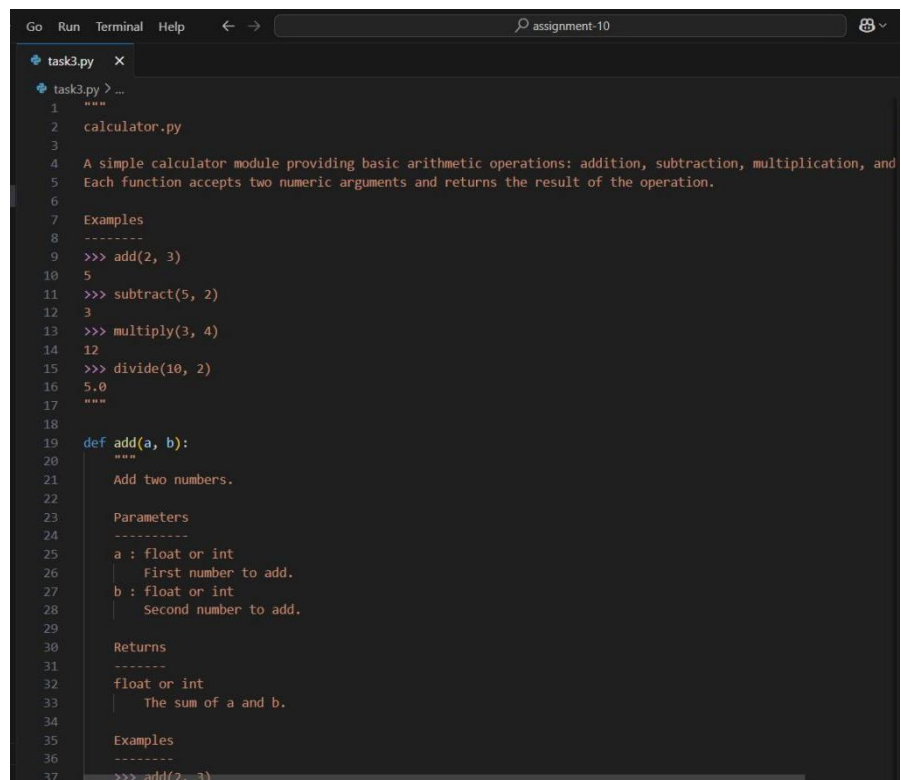
- Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
- Incorporate manual **docstring** in code with NumPy Style
- Use AI assistance to generate a module-level docstring + individual function docstrings.

- Compare the AI-generated docstring with your manually written one.

PROMPT 1:

Generate a Python program that does basic math (like adding, subtracting, multiplying, dividing) and practice writing clear documentation for your code.

CODE GENERATED:



```
Go Run Terminal Help ← → assignment-10
task3.py X
task3.py > ...
1 """
2 calculator.py
3
4 A simple calculator module providing basic arithmetic operations: addition, subtraction, multiplication, and
5 Each function accepts two numeric arguments and returns the result of the operation.
6
7 Examples
8 -----
9 >>> add(2, 3)
10 5
11 >>> subtract(5, 2)
12 3
13 >>> multiply(3, 4)
14 12
15 >>> divide(10, 2)
16 5.0
17 """
18
19 def add(a, b):
20     """
21     Add two numbers.
22
23     Parameters
24     -----
25     a : float or int
26     | First number to add.
27     b : float or int
28     | Second number to add.
29
30     Returns
31     -----
32     float or int
33     | The sum of a and b.
34
35     Examples
36     -----
37     >>> add(2, 3)
```

```
Go Run Terminal Help < -> assignment-10
task3.py x
task3.py > ...
19 def add(a, b):
20     """
21     Add two numbers.
22     Parameters
23     -----
24     a : float or int
25     |   Number to add to.
26     b : float or int
27     |   Number to add.
28     Returns
29     -----
30     float or int
31     |   The result of a plus b.
32     Examples
33     -----
34     >>> add(2, 3)
35     5
36     """
37     return a + b
38
39 def subtract(a, b):
40     """
41     Subtract one number from another.
42     Parameters
43     -----
44     a : float or int
45     |   Number to subtract from.
46     b : float or int
47     |   Number to subtract.
48     Returns
49     -----
50     float or int
51     |   The result of a minus b.
52     Examples
53     -----
54     >>> subtract(5, 2)
55     3
56     """
57     return a - b
58
59 def multiply(a, b):
60     """
61     Multiply two numbers.
62     Parameters
63     -----
64     a : float or int
65     |   First number to multiply.
66     b : float or int
67     |   Second number to multiply.
68     Returns
69     -----
70     float or int
71     |   The product of a and b.
72     Examples
73     -----
74     >>> multiply(3, 4)
75     12
76     """
77     return a * b
78
79 def divide(a, b):
80     """
81     Divide one number by another.
82     Parameters
83     -----
84     a : float or int
85     |   Numerator.
86     b : float or int
87     |   Denominator.
88     Returns
89     -----
90     float
91     |   The result of a divided by b.
92     Raises
93     -----
94     ZeroDivisionError
95     |   If b is zero.
96     """
97     return a / b
98
99 if __name__ == '__main__':
100     # Example usage
101     print(add(2, 3))
102     print(subtract(5, 2))
103     print(multiply(3, 4))
104     print(divide(10, 2))
105
106 # End of file
```

```
Go Run Terminal Help < -> assignment-10
task3.py x
task3.py > ...
65 def multiply(a, b):
66     """
67     Multiply two numbers.
68     Parameters
69     -----
70     a : float or int
71     |   First number to multiply.
72     b : float or int
73     |   Second number to multiply.
74     Returns
75     -----
76     float or int
77     |   The product of a and b.
78     Examples
79     -----
80     >>> multiply(3, 4)
81     12
82     """
83     return a * b
84
85 def divide(a, b):
86     """
87     Divide one number by another.
88     Parameters
89     -----
90     a : float or int
91     |   Numerator.
92     b : float or int
93     |   Denominator.
94     Returns
95     -----
96     float
97     |   The result of a divided by b.
98     Raises
99     -----
100     ZeroDivisionError
101     |   If b is zero.
102     """
103     return a / b
104
105 if __name__ == '__main__':
106     # Example usage
107     print(multiply(3, 4))
108     print(divide(10, 2))
109
110 # End of file
```

```
Go Run Terminal Help ← → assignment-10
task3.py ×
task3.py > ...
88 def divide(a, b):
102     """
103     The result of a divided by b.
104
105     Raises
106     -----
107     ZeroDivisionError
108     If b is zero.
109
110     Examples
111     -----
112     >>> divide(10, 2)
113     5.0
114     """
115     if b == 0:
116         raise ZeroDivisionError("Cannot divide by zero.")
117     return a / b
118
119 def main():
120     print("Addition: add(2, 3) =", add(2, 3))
121     print("Subtraction: subtract(5, 2) =", subtract(5, 2))
122     print("Multiplication: multiply(3, 4) =", multiply(3, 4))
123     print("Division: divide(10, 2) =", divide(10, 2))
124     # Example for division by zero
125     try:
126         print("Division: divide(5, 0) =", divide(5, 0))
127     except ZeroDivisionError as e:
128         print("Division: divide(5, 0) raised an error:", e)
129
130 if __name__ == "__main__":
131     main()
```

OUTPUT:

```
wdownloads/AI ASSISTED CODING/assignment-10/task3.py"
Addition: add(2, 3) = 5
Subtraction: subtract(5, 2) = 3
Multiplication: multiply(3, 4) = 12
Division: divide(10, 2) = 5.0
Division: divide(5, 0) raised an error: Cannot divide by zero.
```

OBSERVATION:

This assignment combines Python function design with good documentation practices. It reinforces core concepts like functions and parameters while promoting clear, professional NumPy-style docstrings. Comparing manual and AI-generated docstrings improved understanding of clarity and formatting.